Ammendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended) A computer-implemented method ~~to emulate~~ for software emulation of a cellular automata based random number generator (CA-based RNG) ~~in software~~, comprising:

determining a set of emulation parameters for ~~said~~ the CA-based RNG;

initializing ~~said~~ the software emulation according to the emulation parameters;

~~simulating behaviors of cells of said CA-based RNG in parallel; and~~

storing state values from odd-numbered cells in a first software variable and state values from even-numbered cells in a second software variable wherein word operations on the first and second variables enable the simulation of the cells to occur in parallel when executed; and

outputting ~~result of said simulation step~~ a random number having the state values stored in the first software variable and the second software variable.

2. (currently amended) The computer-implemented method of claim **1** wherein [[said]] the set of emulation parameters include at least one of a seed, a number of runs[[,]] a cell truth table, interconnection displacement values, and an output designation.

3. (currently amended) The computer-implemented method of claim **1** wherein ~~said step of~~ determining the set of emulation parameters includes retrieving said emulation parameters from at least one of a user, a file, a database, and a remote source.

4. (currently amended) The computer-implemented method of claim **1** wherein ~~said step of~~ determining the set of emulation parameters includes providing default values for any emulation parameters not received from an external source.

5. (currently amended) The computer-implemented method of claim **1** wherein ~~said simulating step includes~~ storing state values from odd-numbered cells in a first software variable and state values from even-numbered cells in a second software variable effects parallel site spacing.

6. (currently amended) The computer-implemented method of claim **1** wherein [[said]] outputting [[step,]] a random number comprises destinations that include at least one of a display, a file, a database, an application, and a remote destination.

7. (currently amended) A computer-implemented method to generate a software code ~~which emulates~~ emulating a cellular automata based random number generator (CA-based RNG), ~~said method~~ comprising:

determining RNG parameters;

determining one or more programming language templates;

determining functional definition of [[said]] the CA-based RNG;

determining initialization routines for [[said]] the CA-based RNG;

determining simulation routines for [[said]] the CA-based RNG;

determining simulation results destination routines for [[said]] the CA-based RNG; and

outputting code for [[said]] the CA-based RNG.

8. (currently amended) The computer-implemented method of claim 7, wherein [[said]] the RNG parameters include at least one of an interconnection topology, a length of [[said]] the desired CA-based RNG, a cellular automata truth table, one or more desired output programming language languages, one or more code output destinations, and site spacing specification.

9. (currently amended) The computer-implemented method of claim 7, wherein ~~said step of~~ determining RNG parameters includes retrieving [[said]] emulation parameters from at least one of one of a user, a file, a database, and a remote source.

10. (currently amended) The computer-implemented method of claim 7, wherein ~~said step of~~ determining RNG ~~emulation~~ parameters includes providing default values for any ~~emulation~~ parameters not received from an external source.

11. (currently amended) The computer-implemented method of claim 7, wherein ~~said step of~~ determining programming language templates includes retrieving [[said]] programming language templates from at least one of one of a user, a file, a database, and a remote source.

12. (currently amended) The computer-implemented method of claim 7, wherein ~~said step of~~ determining programming language templates includes providing at least one default language template in response to no language being specified from an external source.

13. (currently amended) The computer-implemented method of claim 7, wherein ~~said step of~~ determining programming language templates includes providing at least one built-in language template in response to no language being specified from an external source.

14. (currently amended) The computer-implemented method of claim 7, wherein ~~said step of~~ determining functional definition includes generating a Boolean sum-of-products equation for [[said]] the CA-based RNG.

15. (currently amended) The computer-implemented method of claim 7, wherein ~~said step of~~ determining simulation routines includes generating code for one or both of operations for parallel simulation and operations for parallel site spacing.

16. (currently amended) The computer-implemented method of claim 7, wherein ~~said step of~~ determining simulation result[[s]] destination routines[[,]] includes generating code for destinations of simulation results [[to]] for a set including ~~include~~ at least one of a display, a file, a database, an application, and a remote destination.

17. (currently amended) The computer-implemented method of claim **7**, wherein ~~said step of~~ outputting code[[,]] includes destinations for [[said]] the code ~~include~~ including at least one from a set of a display, a file, a database, a compiler, a remote destination, and an interpreter.

18. (currently amended) A computer system having one or more modules for generating [[a]] software code to emulate a cellular automata based random number generator (CA-based RNG), said computer system comprising:

a RNG-parameter module ~~determining~~ that determines RNG parameters[[;]],

a language-template module ~~determining~~ that determines language templates[[;]], and

code-generating module ~~generating~~ that generates code for emulating said CA-based RNG.

19. (currently amended) The computer system of claim **18**, wherein [[said]] the RNG parameters include at least one of an interconnection topology, a length of ~~said desired~~ the CA-based RNG, a cellular automata truth table, one or more desired output programming language languages, one or more code output destinations, and site spacing specification.

20. (currently amended) The computer system of claim **18**, wherein [[said]] the RNG parameter-module retrieves [[said]] the emulation parameters from at least one of a set including one of a user, a file, a database, and a remote source.

21. (currently amended) The computer system of claim **18**, wherein [[said]] the RNG parameter-module provides default values for ~~any emulation~~ RNG parameters not received from an external source.

22. (currently amended) The computer system of claim **18**, wherein [[said]] the language-template-module retrieves [[said]] the language templates from at least one of one of a set including a user, a file, a database, and a remote source.

23. (currently amended) The computer system of claim **18**, wherein [[said]] the language-template-module provides at least one default language template in response to no language being specified from an external source.

24. (currently amended) The computer system of claim **18**, wherein [[said]] the language-template-module provides at least one built-in language template in response to no language being specified from an external source.

25. (currently amended) The computer system of claim **24**, wherein [[said]] the code-generating-module further comprises ~~includes~~:

a functional-definition-module ~~generating~~ that generates a functional definition of said CA-based RNG;

an initialization-generation-module ~~generating~~ that generates initialization routines;

a simulation-generation module ~~generating~~ that generates simulation routines; and

an output-generation-module ~~generating~~ that generates simulation result[[s]] output routines.

26. (currently amended) The computer system of claim **25**, wherein [[said]] the functional-definition-module [[(532)]] generates a Boolean sum-of-products equation for said CA-based RNG.

27. (currently amended) The computer system of claim **25**, wherein [[said]] the simulation-generation-module [[(536)]] generates code for one or both of operations for parallel simulation and operations for parallel site spacing.

28. (currently amended) The computer system of claim **25**, wherein ~~in said~~ output destinations simulation result output routines generated by [[said]] the output-generation-module include at least one of a set including a display, a file, a database, an application, and a remote destination.

29. (currently amended) The computer system of claim **18**, wherein in [[said]] the code-generating-module outputs [[said]] the generated ~~routines~~ code to at least one of a set including a display, a file, a database, a compiler, a remote destination, and an interpreter.